

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Denis Korinšek

Mobilna aplikacija za zajem in prikaz stereogramov

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2014

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Denis Korinšek

Mobilna aplikacija za zajem in prikaz stereogramov

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTORICA: viš. pred. dr. Alenka Kavčič

Ljubljana, 2014

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuirajo predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V okviru diplomske naloge izdelajte mobilno aplikacijo za platformo Android, ki omogoča zajem stereoskopske slike in njen prikaz v obliki anaglifa. Aplikacija naj omogoča zajem stereoskopske slike na dva načina: sočasno z dvema povezanima napravama ali pa z eno samo napravo, s katero naredimo dva zaporedna posnetka iz dveh različnih zornih kotov. Pri implementaciji rešitve posvetite potrebno pozornost tudi preprostosti in intuitivni uporabi aplikacije.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Denis Korinšek, z vpisno številko **63110055**, sem avtor diplomskega dela z naslovom:

Mobilna aplikacija za zajem in prikaz stereogramov

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom viš. pred. dr. Alenke Kavčič,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 10. septembra, 2014

Podpis avtorja:

Za vso strokovno pomoč, trud in nasvete se iskreno zahvaljujem mentorici viš. pred. dr. Alenki Kavčič. Zahvalil bi se družini za pomoč in podporo ter še posebej mami Albini in očetu Robertu za podporo in potrpežljivost skozi vsa študijska leta. Hvala!

Kazalo

Povzetek

Abstract

Poglavje 1	Uvod	1
Poglavje 2	Uporabljene tehnologije in orodja	3
2.1	Operacijski sistem Android	3
2.2	Android SDK in Eclipse ADT Bundle	4
2.3	Kamera	4
2.4	Razred SurfaceHolder	5
2.5	Tehnologija Bluetooth	5
2.6	Manifest	6
2.7	Photoshop	6
2.8	Opis postavitve mobilne aplikacije	6
2.9	Bitna slika	7
2.10	Razred View	7
2.10.1	Canvas	8
Poglavje 3	Stereoskopija	9
3.1	3D vid	11
3.2	Anaglif	12
3.3	3D Očala	13
3.4	Barvni kanali	14
Poglavje 4	Aplikacija za zajem stereoskopske slike	17
4.1	Načrtovanje	17
4.2	Grafični vmesnik	21
4.3	Urejevalnik slik	22

4.3.1	Obrezovanje slik	24
4.4	Združevanje dveh slik v anaglifske stereoskopske slike	24
4.5	Zajem slike z dvema napravama	25
4.5.1	Zajem slike z eno napravo	26
4.6	Problemi pri implementaciji	27
4.6.1	Pomanjkanje pomnilnika za delo z bitnimi slikami	27
4.6.2	Prenašanje slik preko povezave Bluetooth	27
4.6.3	Prikaz napačno obrnjene slike	28
4.6.4	Shranjevanje slike	28
4.7	Testiranje aplikacije	29
Poglavje 5	Sklepne ugotovitve	31

Seznam uporabljenih kratic

kratica	angleško	slovensko
3D	Three dimensional	tridimenzionalno
XML	Extensible Markup Language	razširljiv označevalni jezik
ADT	Android Development Tools	razvojna orodja za operacijski sistem Android
ARGB	Color model with Alpha, Red, Green and Blue channels	barvni model z alfa, rdečim, zelenim in modrim kanalom
SDK	Software development kit	paket za razvoj programske opreme
SIFT	Scale-Invariant Feature Transform	vizualni opisnik

Povzetek

V diplomskem delu smo predstavili razvoj mobilne aplikacije za operacijski sistem Android, ki s pomočjo pametnih mobilnih naprav omogoča zajemanje 3D slike in prikaz stereoskopske (anaglifske) slike. Namen aplikacije je, da lahko uporabnik zajeme fotografijo v 3D pogledu, katera skozi namensko 3D očala uporabniku omogoča pogled navidezno prostorske slike. Cilj diplomske naloge je bil, da čim boljše raziščemo zajemanje prostorskega posnetka s stereoskopsko tehniko in uporabniku ponudimo kvaliteten in uporaben končni izdelek, ki bo omogočal zajem dveh različnih pogledov glede na okolico. Mobilna aplikacija ima dva načina zajemanja slike, in sicer z eno ali pa dvema mobilnima napravama. V primeru zajema z dvema napravama velik del zajemanja in delovanja aplikacije temelji na komunikaciji med mobilnima napravama. Kadar zajemamo sliko z enim telefonom, pa uporabniku omogočimo karseda enostaven in učinkovit zajem fotografije ter mu ponudimo smernice za kvaliteten zajem.

Ključne besede: anaglif, stereoskopija, Android

Abstract

In the diploma work we presented the development of a mobile application for the operation system Android which with the help of smart mobile devices enables the capture of 3D images and the presentation of stereoscopic image. The purpose of the application is that the user can capture an image in the 3D view which with the help of 3D glasses enables the user to see seemingly spatial images. The goal of the diploma work is to research capturing such images with the stereoscopic technology and offer the user a high quality product which will enable two different views of the surroundings. The mobile application has two different ways of capturing images, either with one or two devices. In the instance of using two devices the large part of capturing the image is based on the communication between the mobile devices. When an image is captured with one device the user is able to capture the image in a simple and effective way and we also offer him guidelines for high quality capturing.

Keywords: anaglyph, stereoscopy, Android

Poglavje 1 Uvod

Med uporabniki je vedno bolj priljubljen prostorski pogled na sliko, ki daje občutek večdimenzionalnosti. V diplomski nalogi smo raziskali možnosti za zajem 3D slike in prikaz anaglifa s pomočjo mobilnega telefona. Takšen način zajemanja je uporabniku zanimiv, ker lahko samostojno brez posebnih kamer in fotoaparátov ustvari 3D sliko, ki mu skozi namenska očala prikaže prostorski pogled na objekt.

Za raziskovanje in delo s 3D sliko smo se odločili, ker nas zanima prostorski pogled in ker želimo ponuditi mobilno aplikacijo, kjer lahko s kamero mobilne naprave brez dodatnih pripomočkov zajemamo in prikazujemo prostorsko sliko. Takšno mobilno aplikacijo lahko uporabimo za zajem okolice ali objekta, ki ga želimo prikazati z učinkom dodatne dimenzije, ki daje uporabniku občutek globinskega pogleda.

Prva stereoskopska slika je bila narejena leta 1840, vendar tisti čas niso prišle do izraza in so se uveljavile šele okoli leta 1930, ko se je začela stereoskopija uporabljati za filme in revije. Leta 1849 je David Brewster iznašel napravo za gledanje stereogramov, ki je bila sestavljena iz zaprte škatle, ki je z uporabo leč iz dveh dvodimenzionalnih slik ustvarila navidezen tridimenzionalen pogled. V tistem času so bile stereoskopske slike priljubljene v sobah za zabave, kjer so imeli izobešene okvirjene slike. Danes se 3D pogled večinoma uporablja v filmski industriji, revijah in fotografijah.

3D slike lahko gledamo tudi s prikazovalniki v obliki čelade [6], ki si jih pritrdimo na glavo. Naprava prikazuje sliko na dveh prikazovalnikih v stilu očal, kjer se prikazuje stereoskopska vsebina. Prikazovalnika sta iz tekočih kristalov, pri čemer se na vsak prikazovalnik prikaže ustrezna slika stereograma ter s tem ustvari navidezno 3D sliko. Moderna naprava Oculus Rift predstavlja sodobni naglavni prikazovalnik, ki z novimi tehnologijami daje občutek globinskega vida. Prikazovalnik vsebuje senzorje, ki z zaznavanjem zunanjega sveta ustvarijo realen pogled, ki je primeren tudi za bolj zahtevna prikazovanja, kot so računalniške igre.

Nekateri televizorji in kinematografi za prikaz uporabljajo aktivno sliko, potrebujemo aktivna 3D očala. Takšna očala delujejo v sinhronizaciji s projekcijsko napravo, kjer očala izmenično blokirajo vid in tako naprava izmenično vsakemu očesu prikaže sliko ter tako ustvari učinek

3D vida. V diplomskem delu bomo uporabljali pasivna 3D očala, ki jih podrobneje predstavimo v poglavju 3.3. Pri pasivnih 3D očalih prikazovalnik vseskozi prikazuje dvojno sliko.

V naslednjem poglavju je predstavljena programska oprema, uporabljene tehnologije in strojna oprema, ki smo jo uporabili pri implementaciji naše rešitve.

V poglavju 4 podrobneje predstavimo, kako deluje naša aplikacija, in opišemo implementacijo ter pristop k rešitvi. Najprej predstavimo zasnovo mobilne aplikacije v fazi načrtovanja, kjer podrobneje opišemo funkcionalnosti, ki jih aplikacija podpira. Funkcionalnosti so predstavljene z opisom delovanja, načinom izvedbe in nekatera tudi dodatno podkrepljena s slikovnim gradivom. Pokažemo možnosti za prikazovanje anaglifa in zajemanje stereogramov s pomočjo dveh mobilnih naprav, kar predstavlja večjo natančnost zajemanja dveh prostorskih pogledov, ki daje boljši občutek za prilagajanje razdalje med napravama in izboljša prikaz prostorskega pogleda. Opišemo težave in probleme, s katerimi smo se srečali pri razvoju mobilne aplikacije. Predstavimo način testiranja mobilne aplikacije, kjer testne naprave med seboj tudi primerjamo ter pokažemo pokritost celotnega spektra kvalitete mobilnih naprav.

V Googlovi spletni trgovini Play je na voljo kar nekaj aplikacij, ki se ukvarjajo z zajemom stereoskopskih slik, zato je naš cilj uporabnika prepričati, da je naša mobilna aplikacija boljša, kakovostnejša in primernejša. V fazi načrtovanja smo veliko časa porabili za dober načrt in definicijo specifikacij, kar posledično predstavlja boljšo in kvalitetnejšo mobilno aplikacijo.

Poglavje 2 Uporabljene tehnologije in orodja

2.1 Operacijski sistem Android

Operacijski sistem Android [2] je razvilo podjetje Google in je zasnovan na jedru Linux. Je platforma, ki daje razvijalcem možnost razvijanja aplikacij in iger za mobilne naprave. Primarno je bil razvit za naprave z zaslonom na dotik, kot so pametni telefoni in tablični računalniki. Zaradi velikega napredka tehnologije se danes uporablja tudi za televizije, avtomobile, ure, igralne konzole, digitalne kamere in druge naprave. Podjetje Google ga ponuja popolnoma brezplačno s popolno izvirno kodo. Leta 2008 je Google izdelal trgovino Android, ki za uporabnika predstavlja poenostavljen prenos mobilnih aplikacij neposredno na mobilno napravo. Logotip Androida (slika 2.1) je enostaven, prepoznaven in predstavlja robota.



Slika 2.1: Logotip operacijskega sistema Android.

Za razvoj mobilne aplikacije za operacijski sistem Android smo se odločili, ker trenutno pokriva najširšo množico uporabnikov. Je najbolj priljubljen operacijski sistem za mobilne naprave, saj po zadnjih analizah [20] mednarodne organizacije IDC zaseda 84,7 % trga. Ker ima skoraj vsaka pametna naprava z operacijskim sistemom Android v strojni opremi kamero ter brezžično tehnologijo Bluetooth, zelo razširimo krog uporabnikov in lahko našo aplikacijo namestimo na večino mobilnih naprav. Ima zelo dobro podporo za razvijalce in veliko uporabne literature ter spletnih strani, ki pomagajo pri razvoju. Da lahko napišemo mobilno aplikacijo, potrebujemo Android SDK in urejevalnik za pisanje programske kode.

2.2 Android SDK in Eclipse ADT Bundle

Android SDK [10] vsebuje množico razvojnih orodij, ki vključujejo razhroščevalnik, knjižice, emulator osnovan na brezplačnem odprtokodnem projektu QEMU, dokumentacijo, primere ter vodiče. Podprti operacijski sistemi na katerih ga lahko uporabljamo, so Linux, Mac OS in Windows. Uradno podprta integrirana razvojna okolja za Android SDK so Eclipse z vtičnikom Android ADT, IntelliJ IDEA in NetBeans z vtičnikom. Uveljavljen standardni format za prevedene aplikacije je .apk. Veliko knjižnic, ki so sestavni del Android SDK, smo uporabili tudi pri izdelavi diplomske naloge. Uporabili smo razrede za uporabo in delo s kamero, shranjevanje bitnih slik, izrisovanje in postavitev uporabniškega vmesnika ter še druge, ki jih v naslednjih poglavjih tudi podrobneje opišemo.

ADT Bundle je integrirano razvojno okolje, ki vključuje vse vtičnike za razvoj mobilnih aplikacij Android. ADT Bundle vsebuje:

- Eclipse z vtičnikom ADT,
- orodja Android SDK,
- platforma Android z orodji,
- najnovejša platforma Android,
- najnovejša različica emulatorja.

Za razvoj z njim smo se odločili, ker razvijalcem ponuja vse potrebne stvari za razvoj v eni namestitvi in se razvijalcu ni treba ukvarjati z nameščanjem vseh potrebnih razvojnih orodij. Ponuja bogato razvojno okolje, ki vsebuje vse funkcionalnosti Eclipsea, s tem pa pridobimo zelo dober urejevalnik kode.

2.3 Kamera

Ogrodje Android vključuje tudi podporo različnih kamer in funkcij, ki so na voljo na mobilnih napravah. Za funkcije kamere uporabljamo obstoječ razred Camera [13], s katerim kontroliramo nastavitve ter delovanje. Preden začnemo razvijati našo aplikacijo z vmesnikom za kamero, moramo v datoteko manifest vključiti dovoljenja za uporabo kamere.

Kamera je za nas ena glavnih mobilnih komponent, ki jih potrebujemo za delovanje aplikacije. Ker za zajem stereoskopske slike potrebujemo fotoaparatus, smo pri aplikaciji uporabljali komponento telefona, ki je integrirana že skoraj v vsaki mobilni napravi z operacijskim sistemom Android. Služi za zajemanje fotografije z eno ali dvema mobilnima napravama, kjer prikazujemo pogled kamere s pomočjo razreda SurfaceHolder, ki poskrbi za prikaz slike v aplikaciji.

2.4 Razred SurfaceHolder

Ker smo morali ponuditi pogled kamere v aplikaciji, smo uporabili razred SurfaceHolder, ki poskrbi, da se prikazuje slika kamere v realnem času v naši aplikaciji. Takšna funkcionalnost je nujno potrebna, da uporabniku prikažemo pogled zajemanja in izboljšamo natančnost zajete fotografije. Na nivoju razreda SurfaceHolder smo poskrbeli, da se kamera zažene in se nastavijo vsi parametri, potrebni za pravilno delovanje in prikaz pogleda. Z njim smo razširili naša razreda za zajemanje z enim in z dvema mobilnima napravama, ki vključujeta prikazovanje pogleda kamere. Ob ustvarjanju prikazovalne površine poskrbimo za obračanje kamere, ker je pogled prevzeto prikazan z napačno obrnjenostjo; problem je opisan v poglavju 4.7.3.

2.5 Tehnologija Bluetooth

Raziskovalci podjetja Ericsson so se začeli s to tehnologijo ukvarjati leta 1998 z namenom, da bi brezžično povezali mobilni telefon s slušalkami. Podjetja, ki promovirajo in uporabljajo to tehnologijo, so združena v posebno interesno združenje Bluetooth Special Interest Group.

Bluetooth [1] je tehnologija za brezžično povezovanje elektronskih naprav. Moč elektromagnetnega valovanja 2,4GHz je majhna, zato je doseg pošiljanja možen le na kratke razdalje. Bluetooth ima doseg okoli 10 metrov in je primeren za prenos majhnih in srednje velikih datotek, saj se podatki v primerjavi z drugimi tehnologijami prenašajo srednje hitro. Zaradi nizke porabe električne energije je zelo primeren za majhne električne naprave, ki za napajanje uporabljajo vgrajene baterije, kot so mobilni telefoni ali pa tablični računalniki.

Zaradi dostopnosti na mobilnih napravah in prenosne hitrosti je tehnologija Bluetooth primerna za komunikacijo in prenašanje podatkov v naši mobilni aplikaciji. Ker pri zajemanju fotografij z dvema mobilnima napravama potrebujemo razmik med napravama, ki je približno 6,5 cm, nam doseg ne predstavlja problema.

Pri delu s tehnologijo Bluetooth moramo poskrbeti tudi za ustrezna dovoljenja v datoteki manifest.

2.6 Manifest

Vsaka aplikacija Android mora v glavnem direktoriju vsebovati datoteko `AndroidManifest.xml`. Datoteka manifest [12] vsebuje glavne informacije o aplikaciji, da operacijski sistem Android lahko zažene ustrezno kodo aplikacije.

Datoteka manifest služi za:

- poimenovanje Java paketa aplikacije,
- opisuje komponente aplikacije (aktivnosti, imena razredov ...),
- določa, katere pravice aplikacije morajo biti dovoljene, da lahko dostopamo do zaščitnih delov in uporabljamo strojno opremo,
- določa najmanjšo različico sistema Android, ki jo aplikacija zahteva za delovanje,
- skrbi za povezavo s knjižnicami, ki jih aplikacija potrebuje za delovanje.

2.7 Photoshop

Program Photoshop smo uporabili za testiranje nastavitv stereoskopske slike, saj smo pred samim razvojem združevanja dveh pogledov morali ugotoviti, kakšne operacije in nastavitve moramo izvršiti nad slikami obeh pogledov. Ker je Photoshop zelo močno in kompleksno orodje, smo lahko testirali vidljivost ter združevali različne slike iz različnih pogledov. Z nastavitvami prosojnosti ter izključevanjem kanalov na sliki smo tudi sestavili prvo testno anaglifsko sliko ter tako ugotovili zaporedje operacij, ki jih moramo uporabiti za pravilno sestavljanje prostorskega pogleda.

Photoshop je grafični urejevalnik, ki ga je razvilo podjetje Adobe Systems za operacijska sistema Windows in OS X. Photoshop podpira večino grafičnih formatov in poleg ostalih uporablja tudi svoja formata PSD in PSB. Uporabljamo ga lahko tudi za pisanje in urejanje tekstov na slikah, izdelavo vektorskih grafik, 3D grafike in med drugim tudi za urejanje video vsebin.

2.8 Opis postavitve mobilne aplikacije

Postavitev [16] definira vizualno strukturo za uporabniški vmesnik ali postavitev gradnikov v aplikaciji. Postavitev lahko opišemo z deklaracijo elementov uporabniškega vmesnika v datoteki XML ali pa s postavljanjem in urejanjem postavitve v realnem času. Android SDK ponuja prilagodljivost za uporabo ene ali obeh metod za deklariranje in urejanje uporabniškega vmesnika aplikacije.

Ker je oblikovanje mobilne aplikacije zelo pomembno za pozitivno uporabniško izkušnjo, smo za oblikovanje oziroma razporeditev elementov aplikacije uporabljali postavitve. Z njimi smo postavili elemente na svoja mesta po v naprej zamišljeni razporeditvi in izgledu aplikacije.

2.9 Bitna slika

Razred Bitmap [9] predstavlja zapis slike, ki ga uporabimo za pošiljanje in obdelovanje bitne slike. Da ustvarimo bitno sliko iz fotografije, uporabimo za to namenjene metode razreda BitmapFactory. Pri uporabi razreda BitmapFactory obstajajo trije načini za izdelavo bitne slike, in sicer iz dodanih slikovnih virov, iz datoteke ali s pomočjo razreda InputStream. Najpogosteje uporabljene metode, s katerimi pretvarjamo sliko, so naslednje:

- za pretvorbo iz virov v bitno sliko se uporablja metoda `decodeResource()`,
- metoda, ki pretvarjata iz datoteke, je `decodeFile()`,
- metoda za pretvarjanje vhodnega toka `InputStreaa` pa je `decodeStream()`.

Težava pri hranjenju bitnih slik je v tem, da lahko pri uporabi velikih slik in fotografij dobimo nekontrolirane izjeme, ki jih je težko obvladovati in nadzorovati, saj datoteke zasedejo veliko prostora in tako lahko zapolnimo ves pomnilnik, namenjen shranjevanju bitnih slik. Različni telefoni imajo različno velik pomnilnik za hranjenje bitnih slik in tako lahko na telefonih z manjšim pomnilnikom pride do izjeme. Problem je podrobneje opisan in razčlenjen v poglavju 4.7.1.

2.10 Razred View

Razred View [11] je osnovni graditveni blok za gradnike uporabniškega vmesnika. Zajema prostor na ekranu in je odgovoren za izrisovanje in lovljenje premikov z miško, pritiske tipk in drugih. Razred View je odgovoren za izrisovanje in prikaz gradnikov, kot so gumbi, vnosna polja, izbirna polja in drugi prikazovalniki, ki se uporabljajo za izdelavo interaktivnih uporabniških vmesnikov.

Razred ViewGroup je razred, v katerem lahko združujemo več pogledov, in je osnovni razred za postavitve v aplikacij. Razdeljen je na nevidne vsebovalnike, ki lahko vsebujejo druge postavitve ali druge skupine postavitev, ki definirajo njihove značilnosti in prikazovalne lastnosti.

2.10.1 Canvas

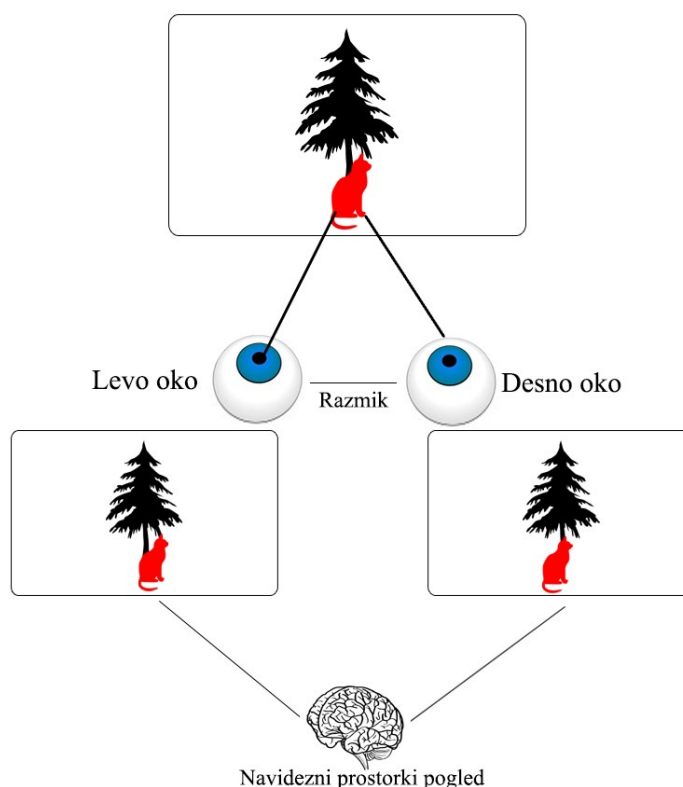
Kadar v mobilni aplikaciji želimo posebno risanje ali prikaz animacije, je pravi način implementacije s pomočjo razreda Canvas. Je risalna površina, kjer se izrisujejo naši ukazi za izris grafičnih likov. V razredu Canvas je risanje izvedeno na osnovi bitnih slik, ki so postavljene na okno zaslona, kamor se izrisuje. Ko definiramo objekt Canvas, moramo kot parameter dodati tudi instanco razreda Bitmap, ki jo uporabimo za izris grafike.

V aplikaciji smo naleteli na problem, ko smo želeli shraniti celotno sliko iz razreda Bitmap, pri kateri bodo ohranjene vse nastavitve transparentnosti. Problem nastane, ker na objekt Canvas dodajamo več različnih bitnih slik. Problem smo rešili s shranjevanjem pomnilnika risanja in tako pridobili vse nastavitve in prikaze iz objekta Canvas v eno sliko. Podrobnejši opis problema je razčlenjen v poglavju 4.7.4.

Ko smo izrisovali bitne slike v objekt Canvas, smo morali poskrbeti tudi za plasti in prosojnost na fotografijah. Prosojnost na slikah smo dosegli tako, da smo uporabili nastavitve alfa kanala, ki predstavlja vrednosti od 0 do 255, kjer minimalna vrednost pomeni popolno prosojnost in maksimalna vrednost neprosojnost fotografije. Za levo sliko smo uporabili kanal alfa z vrednostjo 224 in tako poskrbeli za svetlejšo končno sliko. Za našo desno sliko smo uporabili polovično prosojnost, kar pomeni, da smo nastavili kanal alfa z vrednostjo 128 v razredu Paint, ki poskrbi, da se v naš urejevalnik izriše slika s prosojnostjo. Razred Paint se uporablja za določanje barve in oblikovanja za bitne slike, like in pisave. Z njim lahko poljubno spreminjamo informacije o oblikovanju v realnem času, kar nam omogoča, da lahko z eno instanco razreda oblikujemo več različnih objektov, ki jih izrisujemo.

Poglavje 3 Stereoskopija

Stereoskopija [3] je tehnika prikaza prostorske slike, ki jo sestavimo iz para komplementarnih fotografij, posnetih iz dveh položajev, ločenih z razdaljo, ki je relativna razdalji med očmi ustvari 3D sliko. Če gledamo fotografijo skozi pripravo za gledanje stereogramov, se fotografiji različnih pogledov združita ter prikažeta vizualni občutek dodatne dimenzije globine (slika 3.1), ki ga pri pogledu na samostojno fotografijo ne dobimo. Razvoj stereoskopije nam je pomagal pri razumevanju binokularne paralakse, ki predstavlja razliko postavitve slike levega in desnega očesa, zaradi katere možgani dobijo informacijo za predstavitev 3D slike.



Slika 3.1: Možgani s pomočjo oči ustvarjajo iluzijo prostorskega pogleda.

3D tehnika stereoskopije [14] je tehnika obdelave, ki omogoča pogled navidezne prostorske slike. Deluje tako, da ob uporabi posebnih očal, ali pravilnega načina pogleda, ustvari iluzijo

prostorskega pogleda. Stereoskopija se deli na dva tipa, in sicer na aktivno in pasivno stereoskopijo. Aktivna stereoskopija je prikaz fotografije, kjer prikazovalnik komunicira z očali. Večina aktivnih 3D televizorjev uporablja posebna LCD očala, ki so napolnjena s tekočimi kristali in v lečah blokirajo sliko vsakemu očesu tudi do 120-krat na sekundo. Takšna očala so navadno na baterije, ali pa potrebujejo polnjenje. Pasivna tehnika stereoskopije obstaja že okoli 100 let, vendar se njena uporaba takrat še ni tako uveljavila. Pasivna 3D očala, ki za svoje delovanje ne potrebujejo baterij, se delijo na anaglifska in polarizirana.

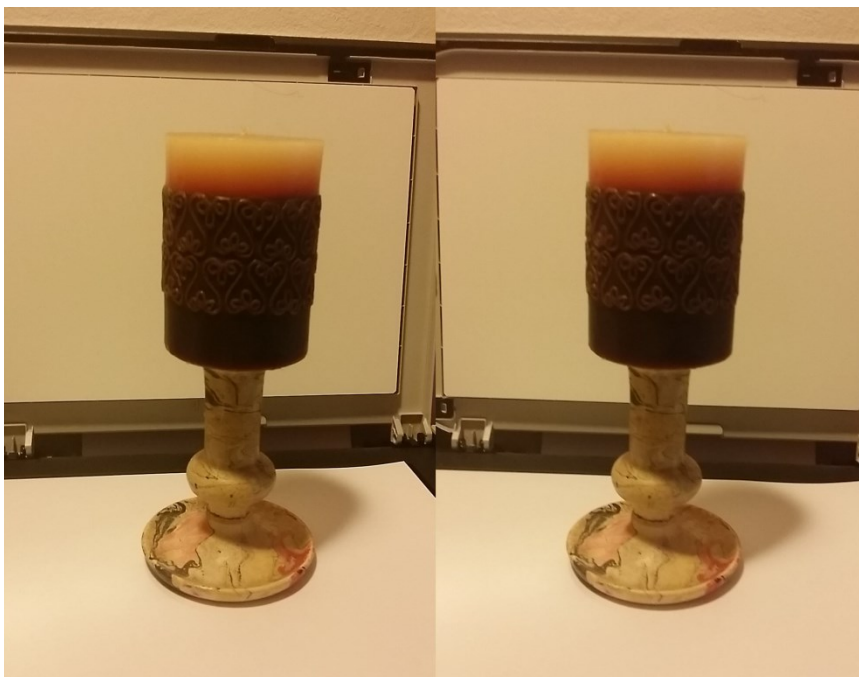
Za zajem kakovostne fotografije s stereoskopskim efektom moramo fotografiji pogleda zajeti z optimalno razdaljo razmika med centroma leč kamer. Razdalja, ki nam prikaže najboljši stereoskopski pogled, je nekje v razdalji človeških oči, ki predstavlja povprečno 64 mm pri odraslem človeku. Naravna paralaksa za dano velikost zaslona predstavlja, kakšen odstotek širine zaslona je enak razdalji človeških oči. Kadar uporabljamo za stereoskopsko širino med fotoaparatom 64 mm in monitor širine 100 cm, moramo širino oči deliti s širino ekrana po enačbi (3.1), da dobimo odstotek za izračun naravne paralakse, kar v našem primeru predstavlja $64/1000 = 0,064$, kar je 6,4 %. To pomeni, da moramo pri televiziji širine enega metra poskrbeti za to, da bo naravna paralaksa največ 6,4 % od širine zaslona.

$$\text{Paralaksa} = (\text{širina med kamerama} \div \text{širina zaslona}) \quad (3.1)$$

Za izračun optimalne razdalje med fotoaparatom se lahko uporabi pravilo 1/30, kar pomeni, da je naša optimalna razdalja približno enaka eni tridesetini razdalje od kamere do najbližjega objekta. Z izračunom pripomoremo k boljši kvaliteti končne slike, kjer pri bližnjih predmetih uporabimo manjšo razdaljo med telefoni. To pravilo velja samo za prikazovalne naprave do velikosti 150 cm, kar je primerno za uporabo pri fotografiranju z našo mobilno aplikacijo. Če želimo izračunati razdaljo med kamerama (B) za najboljši stereoskopski pogled, lahko uporabimo Bercovitzovo formulo (3.2) [15], kjer za izračun uporabimo želeno paralakso (P), najkrajšo (N) in najdaljšo (L) razdaljo od leče kamere in goriščno razdaljo leč (F).

$$B = P \div (L - N) \times (LN \div F - (L + N) \div 2) \quad (3.2)$$

Stereoskopske fotografije lahko gledamo tudi brez očal s pomočjo križnega pogleda. Sliki morata biti postavljeni tako, da je leva slika postavljena na desno stran in desna na levo (slika 3.2). Na tako postavljeno fotografijo gledamo tako, da fokusiramo pogled bližje, kot pa je slika, saj tako gledamo križno in vsako oko vidi samo eno sliko. Takšen način gledanja je utrujajoč za oči.



Slika 3.2: Stereoskopska slika za križni pogled.

3.1 3D vid

V realnem svetu zaznavamo globinski vid zaradi tega, ker imamo dve očesi. Naše levo oko ima nekoliko drugačen pogled kot naše desno oko. Na podlagi različnih pogledov naši možgani sestavijo navidezno globinsko sliko, kjer razlika med slikama predstavlja globino objekta. Takšen sistem vida se je pri človeku razvijal nekaj milijonov let.

Naše oči lahko prikažejo samo dve dimenziji, medtem ko nam globino navidezno predstavijo možgani. Dokaz, da globino res ustvarijo možgani, lahko pokažemo na primeru, ko zapremo eno oko in poskušamo igrati badminton. Zaradi zaprtega očesa vidimo samo dve dimenziji ter težje zaznavamo globino oziroma oddaljenost žogice, zato je ob igranju z enim očesom večja verjetnost za neuspeh.

Če želimo ustvariti 3D računalniški prikaz, moramo ugotoviti, kako prikazati dva različna pogleda našim očem na eni sami sceni. Celotna računalniška 3D grafika se vrti okoli izumljanja pametnih in najboljših načinov za predstavitev ločenih pogledov našim očem.

Metoda, ki jo uporabljamo za izdelavo globinskega pogleda, se imenuje anaglif [8]. Ta metoda temelji na uporabi dveh slik, ki sta združeni v en pogled in prikazani v različnih barvah, ki se med seboj izključujejo.

3.2 Anaglif

Anaglif je ime za stereoskopski 3D efekt, kjer se za vsako oko uporablja filtre drugačnih barvnih odtenkov. Za kodiranje potrebujemo dve komplementarni barvi za filtriranje slik, tipično pa se za levi pogled uporablja rdeč in za desni pogled moder in zelen barvni kanal. Metoda je postala priljubljena šele v zadnjih desetletjih, saj jo lahko prikazujemo na kateremkoli večbarvnem prikazovalniku. Sama metoda ustvarjanja anaglifske 3D slike je enostavna in ne preveč računsko zahtevna, tako da je primerna tudi za delovanje na slabših napravah. Pri pogledu skozi barvna anaglifska očala [7] vsaka od slik zaradi barvnih kanalov doseže svoje oko in tako vidimo navidezno 3D sliko. Vsako oko vidi svojo barvo prav zaradi očal, ki skozi levo rdečo lečo prepuščajo samo moder in zelen kanal ter skozi desno lečo le tiste kanale z rdečo barvo. Primer anaglifske fotografije lahko vidimo na sliki 3.3. Ob gledanju slike skozi barvna očala vizualni korteks možganov združi sliki v percepcijo tridimenzionalnega prostora. To pomeni, da možgani uporabniku ustvarijo iluzijo globinske slike. Anaglifske slike so veliko lažje za prikaz 3D učinka kot paralelni ali križni stereogrami, kjer moramo za prostorski pogled osvojiti posebno tehniko gledanja in premikanja.

Anagliffe [4] lahko kodiramo na več različnih načinov, ki se razlikujejo v uporabi različnih barv na sliki in kvaliteti prikaza prostorskega efekta. Poznamo metode za barvne, črno bele in navadne anagliffe. Metoda navadnih anaglifov daje najboljši 3D pogled, vendar se zmanjša kvaliteta barv in slike. Deluje tako, da uporabimo za desno oko prikaz rdečega kanala ter na sliki levega očesa prikaz modrih in zelenih kanalov. Na enak način deluje črno-bela metoda izdelave anaglifov, kjer ni predstavljena informacija barv. Je najpogostejše uporabljen način izdelave anaglifov, saj je slika običajno boljša za pogled. Metoda barvne anaglifske slike deluje tako, da za kodiranje barv uporablja rumeno za levo sliko in modro-zeleno za desno. Barvni anaglifi poskušajo ohraniti kvaliteto prikaze barve originalne slike, vendar niso vse slike primerne za barvne anagliffe, ker z vsemi očali ne vidimo vseh barv.



Slika 3.3: Anaglifska fotografija, posneta z našo mobilno aplikacijo.

3.3 3D Očala

Obstajajo trije tipi 3D očal, ki se razlikujejo v načinu prikaza slik, ki jih z določenimi očali lahko gledamo. Tipi očal za pogled stereoskopskih slik so: anaglifska, polarizirana in aktivna.

Anaglifska 3D očala [7] so lahko rdeče/modrozeleni, rdeče/modri in rdeče/zeleni in so lahko v plastičnih ali papirnatih okvirjih. Pri izdelavi anaglifske slike smo uporabili rdeč kanal za prikaz desnemu očesu ter moder in zelen za prikaz levemu očesu.

Polarizirana oziroma pasivna 3D očala se pogosto uporabljajo v kinu in za pasivne 3D televizorje. Ta očala uporabljajo polarizirane leče, ki so lahko v navadnih okvirjih za očala.

Aktivna 3D zaklopna očala se uporabljajo za igranje računalniških igr, aktivne 3D televizorje, kjer se LCD leča odpira in zapira v sinhronizaciji s stereo okvirji. Sprejemanje infrardečih signalov poteka tako, da monitor pošilja signale očalom. Za prikaz slike s projektorjem pa se uporabljajo radio-frekvenčni signali.



Slika 3.4: Izgled testnih 3D anaglifskih očal.

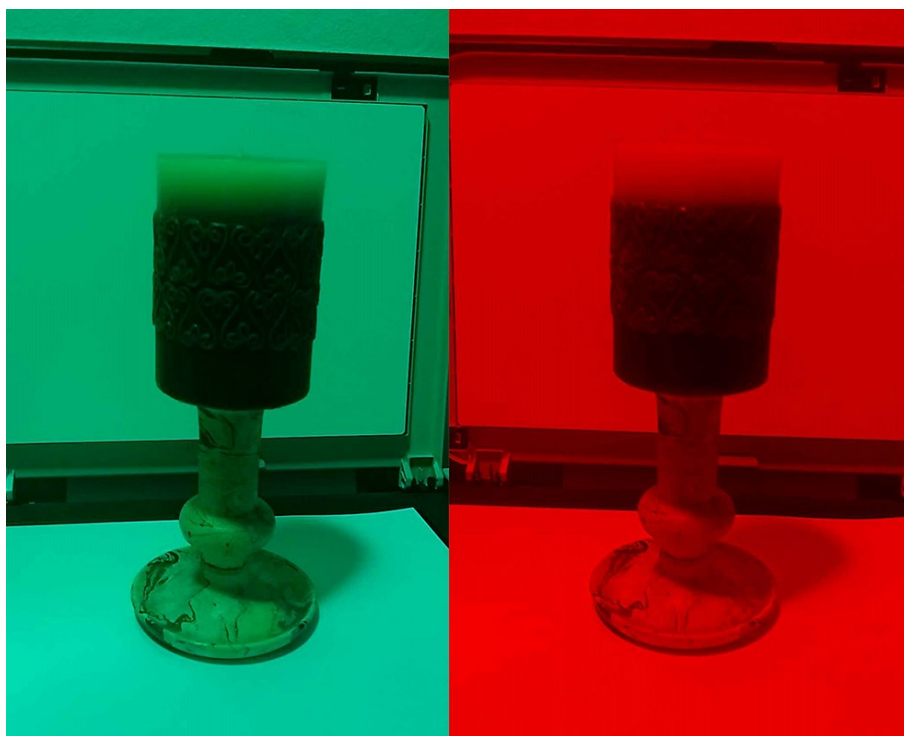
3.4 Barvni kanali

Število kanalov, ki jih ima vsaka slikovna pika ali piksel na sliki, pomeni, s koliko števili predstavimo barvo vsakega piksla. V predstavitvi barvnega modela RGB (rdeča, zelena, modra) ima slika za vsak piksel tri števila, ki predstavljajo njegovo barvo. Na primeru iz prakse to pomeni, da slika, ki ima širino ter višino 1000 pikslov, vsebuje vsega skupaj milijon slikovnih pik. Če za vsak piksel uporabimo po tri števila za zapis barve, vsako število pa zavzame en bajt, bo slika z enim milijonom slikovnih pik zasedla 3 megabajte prostora.

Barvne kanale potrebujemo pri združevanju anaglifskih stereoskopskih slik, saj je zelo pomemben del pri tem ta, da na levi sliki uporabimo samo kanala z modro in zeleno barvo ter na desni sliki z rdečo barvo. Ko sliki združimo, dobimo dva sloja slik, ki sta porazdeljena eden čez drugega, in skozi 3D anaglifska očala vidimo na levo oko samo levo sliko z modro in zeleno barvo ter na desno oko desno sliko z vključenim rdečim kanalom. Takšen način deluje, ker gledamo sliko skozi anaglifska očala, ki za vsako oko uporabljajo drugačen barvni filter. Levo oko gleda na fotografijo skozi rdeče steklo in desno skozi modrozeleno, kar pomeni, da se bo levemu očesu prikazovala samo slika z zelenim in modrim kanalom in desnemu samo slika z rdečim kanalom. Takšen način pogleda možgani sestavijo v navidežno globinski pogled.

V mobilni aplikaciji se ob zagonu urejevalnika slik oba pogleda najprej zakodirata z ustreznimi kanali (slika 3.5). Leva slika je zakodirana z zelenim in modrim kanalom, kar pomeni, da sta na sliki prikazani samo zelena in modra barva, ker je rdeči kanal popolnoma

izklopljen. Na desni sliki pa je ravno obratno in je prikazana samo rdeča barva. Kodiranje deluje tako, da se v zanki sprehodimo čez vse piksele v sliki, kjer barvo na nivoju vsakega piksla preberemo v vrednosti ARGB, ki predstavlja rdečo, zeleno in modro barvo z dodatno informacijo prosojnosti (alfa). V vrednosti ARGB se v primeru vključitve rdečega kanala shranita samo prosojnost in rdeča barvna komponenta piksla, kadar pa kodiramo levo sliko, se poleg prosojnosti zapišeta vrednosti zelene in modre barvne komponente. Ko so nastavljeni ustrezni kanali, se nova barva piksla zapiše nazaj na svoje mesto v bitni sliki. Za gledanje takšne slike potrebujemo očala, ki imajo rdečo in modrozeleno lečo.



Slika 3.5: Fotografiji pogledov, kjer je leva slika zakodirana z zelenim in modrim kanalom ter desna z rdečim kanalom.

Poglavje 4 Aplikacija za zajem stereoskopske slike

Ob zagonu aplikacije ima uporabnik preprost in na pogled prijazen uporabniški vmesnik, ki uporabnika vodi skozi aplikacijo. Kot prva možnost je uporabniku ponujena izbira slikanja stereoskopske slike s pomočjo ene mobilne naprave, kjer uporabnik zajema sliko brez povezave z drugimi telefoni, ampak različna pogleda posname v dveh delih, kjer prvi del predstavlja zajem druge slike, drugi pa nam pri zajemu pomaga z obrisom prve fotografije, da lažje poravnavamo z objekti, posnetimi ob prvem pogledu. Druga možnost zajemanja slik je zajem stereogramov s pomočjo dveh mobilnih naprav, kjer uporabnik za zajem potrebuje dve mobilni napravi, ki se med seboj brezžično povežeta ter sočasno posnameta pogleda iz različnih zornih kotov.

Da uporabnik aplikacije nebi imel problemov z iskanjem zajetih slik oziroma z lokacijo shranjevanja slik, smo izdelali galerijo slik, ki prikaže vse fotografije, zajete z mobilno aplikacijo. Aplikacija slike shranjuje v mapo Steroscope/slike/, kjer se slike samodejno datumsko poimenujejo. Galerija nam prikaže pomanjšane slike, kjer imamo dodano možnost povečevanja oziroma pogleda v celozaslonskem načinu.

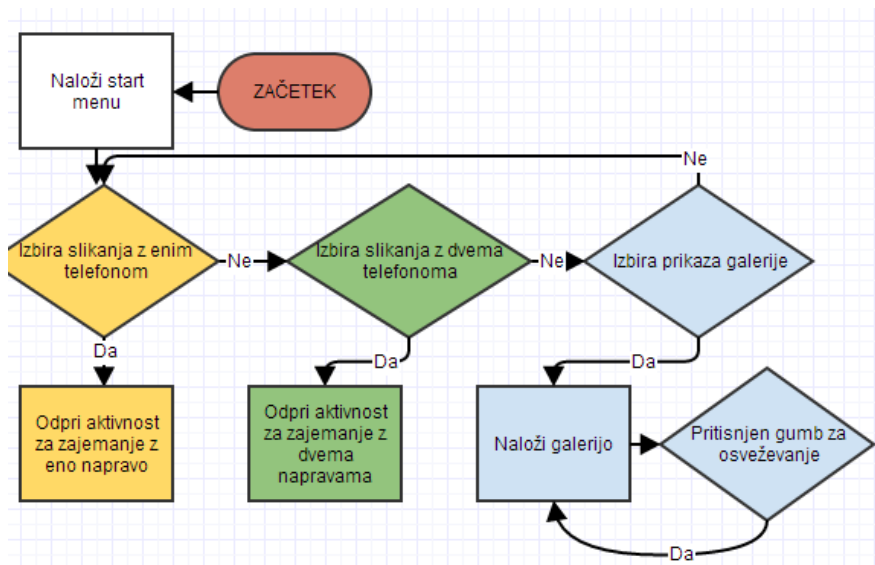
4.1 Načrtovanje

Aplikacijo smo si zamislili tako, da pokriva čim večji del mobilnih naprav in da je pregledna, enostavna za uporabo ter da uporabniku ponuja kar se da dobro izkušnjo ter kvaliteto zajetih slik.

Omejitve iz tehničnega vidika so v fazi načrtovanja pomemben del, da se pravilno odločimo, katere tehnologije in komponente mobilnih naprav bomo uporabili, ter se že v fazi načrtovanja izognemo prekoračitvi časovnega plana in stroškom zaradi uporabe neustrezne tehnologije. Uporaba komunikacije s tehnologijo Bluetooth je za pošiljanje signala prepočasna in bi, iz tehničnega pogleda, potrebovali veliko hitrejšo komunikacijo, da bi lahko stereoskopske fotografije zajeli tudi na premikajočih predmetih. Namesto tehnologije Bluetooth bi lahko uporabili brezžično komunikacijo Wi-Fi, ampak potem smo omejeni na brezžični internet in zmanjšamo možnost uporabe mobilne aplikacije na terenu, ali pa uporabniku povzročimo morebitne dodatne stroške. Za implementacijo smo kljub slabostim uporabili tehnologijo Bluetooth.

Ker mobilna aplikacija vključuje povezovanje s tehnologijo Bluetooth, zajemanje posnetka s kamero, shranjevanje slik na lokalni disk, oblikovanje slik in prikaz galerije, so za pravilno delovanje potrebne minimalne zahteve. Aplikacija za svoje delovanje potrebuje mobilno napravo, ki podpira pošiljanje s tehnologijo Bluetooth, katera je pomembna za komunikacijo med telefonoma in kamero. Kamera je strojna komponenta, brez katere aplikacija ne deluje, saj brez možnosti fotografiranja zajem slik ni možen. Minimalna različica operacijskega sistema Android za delovanje naše aplikacije je 11 (Android 3.0 Honeycomb). Mobilna aplikacija normalno in brez napak deluje na vseh različicah, ki so novejšje od Android 3.0.

Funkcionalnosti, ki jih naša mobilna aplikacija obsega, smo pri načrtovanju razdelili na pet delov, kjer so nekateri med seboj odvisni. Za nekatere funkcionalnosti smo dobili ideje na podlagi pregleda že obstoječih aplikacij. Prvi del je začetni meni, ki je osrednji del mobilne aplikacije, preko katerega dostopamo do drugih aktivnosti, ki so prikazane na sliki diagrama poteka (slika 4.1). Namen začetne aktivnosti je uporabniku ponuditi funkcionalnosti, ki jih aplikacija obsega, ter omogočiti lažjo navigacijo med aktivnostmi, kjer do večine funkcionalnosti dostopamo na enem mestu ter s prijaznim uporabniškim vmesnikom dosežemo boljši prvi vtis uporabnika.



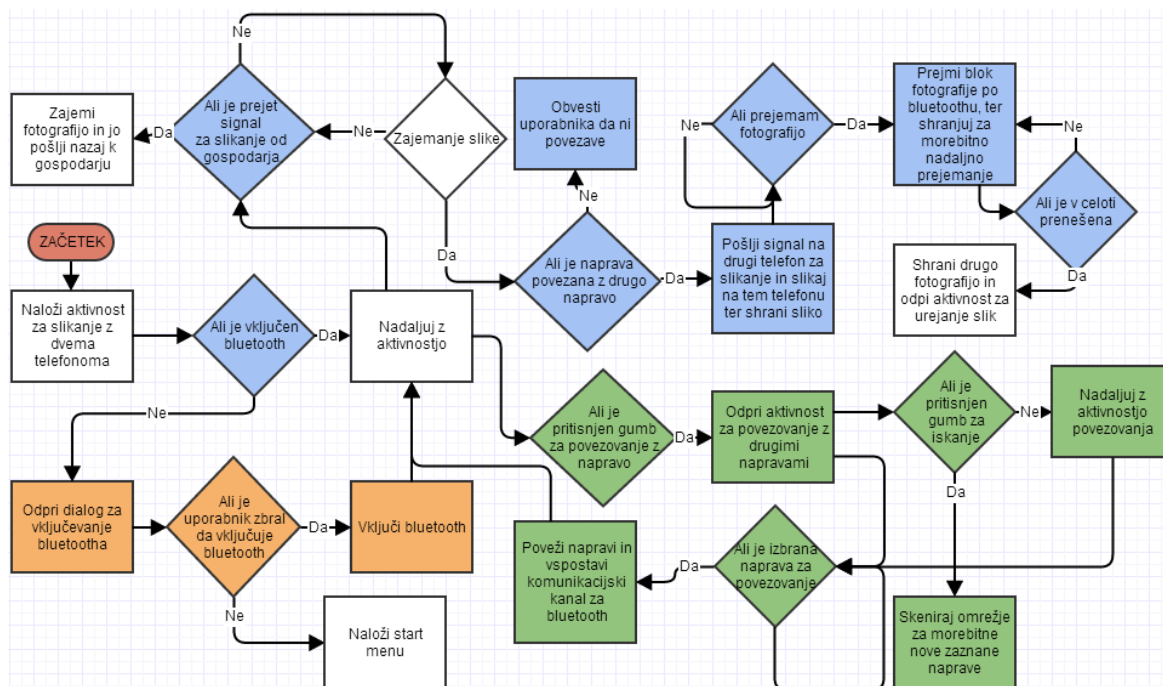
Slika 4.1: Diagram poteka začetne aktivnosti.

Aktivnost za zajemanje stereoskopske fotografije s pomočjo dveh naprav je sestavljena iz dveh implementacijskih delov, kjer v prvem delu sestavimo komunikacijo med mobilnima napravama in pošiljanje fotografij preko protokola Bluetooth, v drugem delu pa smo se posvetili zajemanju in združevanju fotografij v anaglifsko sliko. Na podlagi specifikacij smo izdelali diagram poteka (slika 4.2), ki prikazuje obnašanje in funkcionalnosti za aktivnost

zajemanja z dvema mobilnima napravama. Z modro barvo so označene funkcionalnosti, ki se nanašajo na pošiljanje s tehnologijo Bluetooth, z zeleno pa aktivnost za vzpostavljanje komunikacije z drugo napravo. Oranžna barva označuje aktivnost za pošiljanje podatkov s tehnologijo Bluetooth, ki je nujno potrebna in mora biti izvršena pred modro in zeleno aktivnostjo. V fazi načrtovanja je bilo sprva predvideno združevanje v anaglifsko sliko takoj, ko sta obe fotografiji na glavni napravi, vendar smo v fazi implementacije ugotovili, da je bolje, da naredimo novo aktivnost, ki poskrbi za združevanje in uporabniku omogoči dodatne funkcionalnosti za delo s sliko. Mobilne komponente, kot so kamera, pomnilnik in tehnologija Bluetooth, za delovanje potrebujejo dovoljenje uporabnika za delo ter uporabo njihovih funkcionalnosti, ki so omejene s pravicami. Pravica je omejevanje dostopa do dela kode ali do komponent na mobilni napravi. Omejevanje pravic služi tudi zaščiti kritičnih podatkov in delov kode, ki bi lahko bili zlorabljeni. Vsaka pravica je opredeljena z unikatno oznako. Pogosto oznake kažejo na akcijo oziroma storitev, ki je omejena. Naša mobilna aplikacija potrebuje za dostop do vseh funkcionalnosti dovoljenja uporabnika, zato smo v datoteko manifest dodali potrebna dovoljenja.

Vključiti smo morali pravice za:

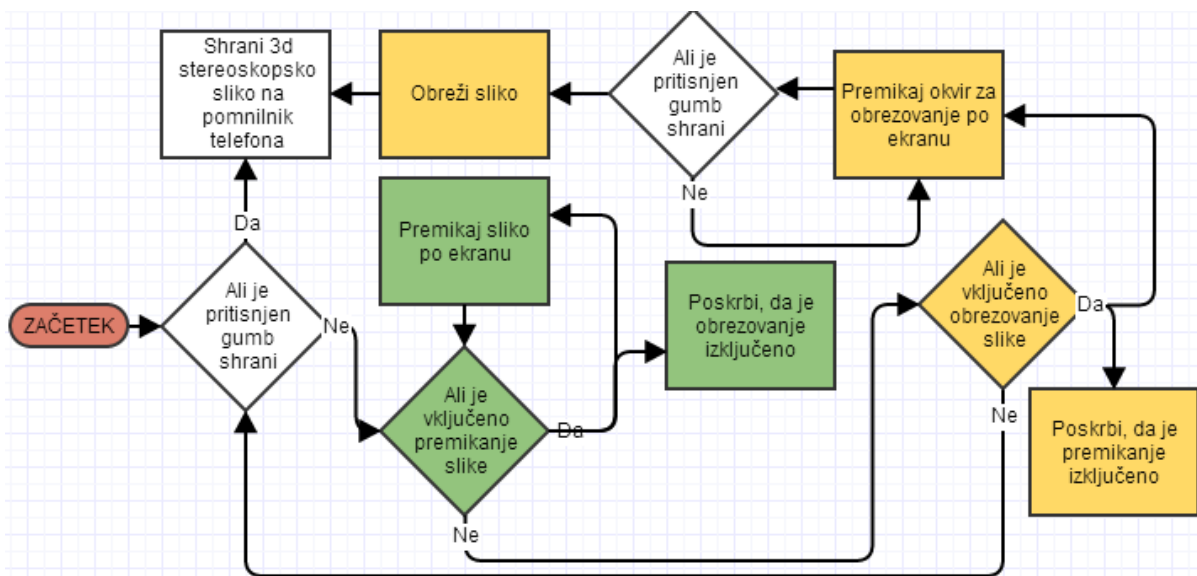
- delo s kamero,
- pisanje v zunanji pomnilnik,
- branje zunanjega pomnilnika,
- vključevanje in upravljanje s tehnologijo Bluetooth,
- pošiljanje in prejemanje podatkov preko tehnologije Bluetooth.



Slika 4.2: Zajemanje stereoskopske slike s pomočjo dveh mobilnih naprav.

Tretja dodatna funkcionalnost je namenjena zajemanju fotografije s pomočjo ene mobilne naprave. Glavni cilj te aktivnosti je zajemanje fotografije z eno napravo, kjer smo ponudili prijazno, logično in učinkovito rešitev. Poglavitna prednost je neodvisnost od več naprav, kjer za zajem ne potrebujemo dodatne opreme, ampak samo našo mobilno napravo. Slabost takšnega zajema se pokaže v primeru, ko bi želeli fotografirati dinamične objekte, ali pa zajeti sliko v gibanju, saj uporabnik zgubi funkcionalnost popolnega samodejnega zajemanja dveh pogledov in mora poglede zajemati ločeno. Ker se temu ne moremo izogniti, predpostavimo, da bo takšno zajemanje namenjeno izključno za fotografiranje statičnih objektov. Izdelavo in urejanje anaglifskih slik pa prepustimo aktivnosti urejevalnika slik.

Urejevalnik slik je četrta aktivnost, ki jo uporabljamo v mobilni aplikaciji. Cilj je bil izdelati učinkovit urejevalnik slik, ki bo poskrbel za združevanje fotografij dveh pogledov v stereoskopsko (anaglifsko) sliko ter pomagal pri urejanju fotografij pred dokončnim shranjevanjem. Glavne funkcionalnosti urejevalnika so prilagajanje, premikanje plasti nad anaglifom in shranjevanje. S premikanjem zgornje plasti anaglifske slike omogočimo, da lahko uporabnik prilagaja navidezno globinsko sliko in paralakso. V fazi implementacije je bila dodana tudi funkcionalnost obrezovanja slike, saj zaradi prilagajanja pride do nepopolnega pokrivanja, kjer lahko odrežemo odvečne robove slike. Aktivnost urejevalnika slik je predstavljena s pomočjo diagrama poteka (slika 4.3), kjer so funkcionalnosti obarvane glede na vrsto. Z zeleno barvo je označeno premikanje fotografije po zaslonu, ki se izključuje z obrezovanjem slike, ki je obarvano z oranžno barvo.



Slika 4.3: Diagram poteka za aktivnost urejevalnika slik.

Galerija slik kot peta funkcionalnost uporabniku omogoča pregled vseh zajetih stereografskih fotografij. Ideja za to funkcionalnost je nastala ob pregledu obstoječe aplikacije 3D Camera. Uporabniku se ob pregledu fotografij prikažejo v obliki pomanjšanih slik, ki se ob izbiri odprejo v celozaslonskem načinu.

4.2 Grafični vmesnik

Ker je sam izgled mobilne aplikacije za uporabnikov prvi vtis zelo pomemben, je ključnega pomena tudi to, kako grafično oblikujemo uporabniški vmesnik. Ker klasični gumbi niso bili primerni za lep izgled naše aplikacije, smo izdelali svojo obliko in stil za gumbe. To smo naredili tako, da smo dodali datoteko tipa XML, kjer smo definirali obliko, kjer se ob pritisku spremenijo barve gumba. Prikaz oblikovanja grafičnega vmesnika za začetni meni je prikazan na sliki 4.4, kjer vidimo oblikovanje gumbov. Na podoben način smo oblikovali tudi ozadje v urejevalniku slik, kjer smo uporabili prelivanje med dvema barvama. Takšen način oblikovanja nam omogoča, da oblikujemo elemente po svoji volji in ustvarimo vizualno privlačno podobo.



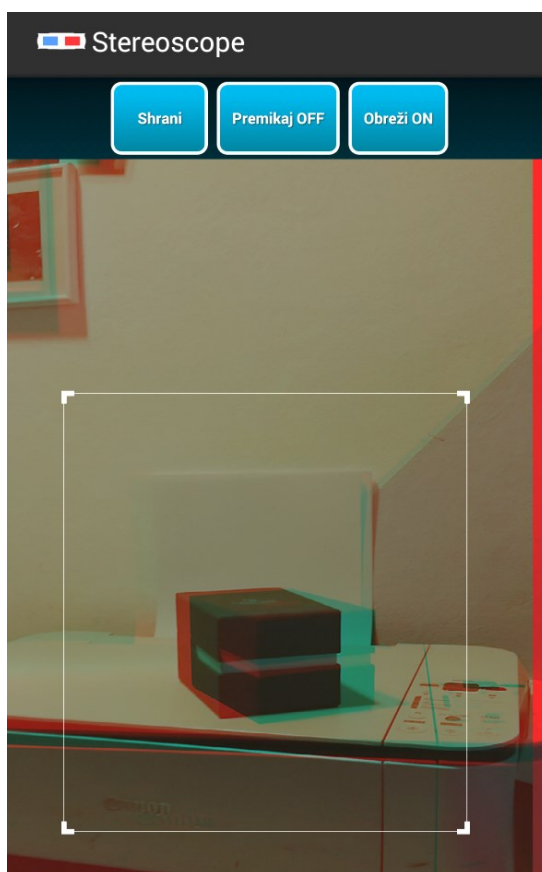
Slika 4.4: Začetni meni, ki prikazuje oblikovanje in postavitev v aplikaciji.

4.3 Urejevalnik slik

Da bi uporabnik lažje obdeloval in prilagajal zajete stereoskopske slike svojemu pogledu, smo naredili temu namenjen urejevalnik slik. Posneti sliki različnih pogledov se samodejno prekrijeta in prikažeta anaglifsko sliko takoj ob zagonu urejevalnika. Urejevalnik, ki je prikazan na sliki 4.5, omogoča ročno prilagajanje prekrivanja zajetih pogledov oziroma kvalitete pogleda za anaglifsko sliko in obrezovanje slike glede na naš pogled. Tako lahko sami prilagajamo kvaliteto ter velikost oziroma izberemo del slike, ki ga želimo uporabiti in prikazati kot končni izdelek. Slike lahko po končanem urejanju in prilagajanju tudi shranimo v pomnilnik telefona. Vse naše shranjene slike pa lahko pogledamo v galeriji.

Vse mobilne naprave Android imajo dva načina shranjevanja, ki sta shranjevanje na notranji in zunanji pomnilnik. Notranji pomnilnik je boljši od zunanjega, kadar želimo do naših datotek dostopati samo mi preko naše aplikacije in s tem omejimo dostop uporabniku ter drugim aplikacijam. Zunanji pomnilnik je primernejši za datoteke, ki ne zahtevajo omejevanje dostopa, in vse datoteke, ki jih želimo deliti z drugimi aplikacijami ali dovoliti dostop drugim

uporabnikov. Slabost zunanjega pomnilnika je ta, da ni vedno dosegljiv. Za notranji in zunanji pomnilnik mora uporabnik aplikaciji dovoliti branje in pisanje. V naši mobilni aplikaciji je za pravilno delovanje nujen zunanji pomnilnik, ker želimo, da uporabnik s svojimi fotografijami poljubno upravlja in jih po potrebi uporabi v drugih aplikacijah za urejanje slik. Prednost tega je tudi, da se slike shranjujejo na ločen medij in tako ob morebitni napaki ali okvari mobilne naprave naše fotografije ostanejo na zunanjem pomnilniku, ki ga lahko brez težav prenesemo v drugo napravo. Uporabljali smo več načinov shranjevanja slik, ker smo na določenih segmentih aplikacije potrebovali shranjevanje bitnih slik v sliko drugega formata, spet drugje smo potrebovali shranjevanje tabele bajtov v sliko ali pa shranjevanje risalnega pomnilnika iz urejevalnika v sliko. Za shranjevanje bitne slike v format JPEG smo uporabili metodo za stiskanje razreda Bitmap, ki poskrbi, da se bitna slika zapiše v izbran format slike ter morebiti zmanjša kvaliteto fotografije, če želimo slike manjših velikosti.

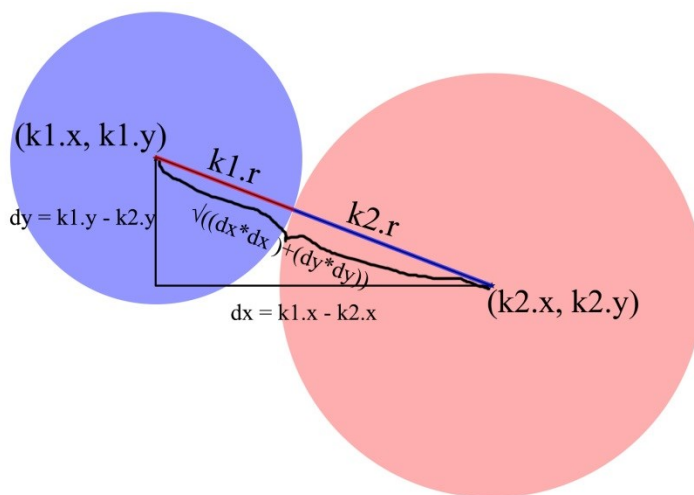


Slika 4.5: Urejevalnik slik, kjer lahko ročno uredimo prekrivanje slike in sliko po potrebi obrežemo, kar se vidi tudi na sliki, kjer beli robovi prikazujejo območje rezanja.

4.3.1 Obrezovanje slik

Stereoskopsko sliko lahko uporabnik končno po svoji volji prilagaja in izboljšuje. S tem dobi kar se da najboljši prostorski pogled, imeti pa mora tudi možnost obrezovanja slik. Obrezovanje potrebujemo, kadar se zaradi premikanja in poravnavanja fotografij sliki ne prekrivata v celoti, ali kadar želi uporabnik izpostaviti ali prikazati samo določene dele fotografije. Robovi, kjer se slika odreže, se v urejevalniku prikazujejo kot pravokotnik, ki daje uporabniku zaradi standardnega izgleda iz drugih slikovnih urejevalnikov informacijo obrezovanja. Ker nimamo na voljo že obstoječega urejevalnika, ki bi omogočal obrezovanje smo implementacijo le-tega sprogramirali sami.

Definirali smo koordinate pravokotnika, ki predstavljajo robove slike, kjer se ob shranjevanju slika obreže. Preverjanje dotika (slika 4.6) za premikanje stranic polja se preverja med točko dotika ($k1.x, k2.y$) in koordinatami oglišč pravokotnika ($k2.x, k2.y$), s pomočjo Pitagorovega izreka, kjer središče kroga ($k1$) predstavlja točko dotika in središče drugega kroga ($k2$) točko oglišča, kjer se slika odreže. Zaradi nenatančnosti našega dotika s prstom, moramo upoštevati okolico dotika, ki je predstavljena z radijem kroga (r) in predstavlja povprečno velikost prsta.



Slika 4.6 Izračun za zaznavanje dotika.

4.4 Združevanje dveh slik v anaglifske stereoskopsko sliko

Ker je glavni namen mobilne aplikacije ustvarjanje stereoskopske (anaglifske) slike, smo se morali posvetiti združevanju slik dveh pogledov, kjer se bodo slike združevale neodvisno od tega, ali gre za zajemanje z eno ali pa z dvema napravama. Združevanje je implementirano tako, da se sliki prebereta iz pomnilnika ter filtrirata s pomočjo metode za prikazovanje kanalov, kjer pri levi sliki prikažemo rdeč kanal in pri desni sliki druga dva kanala brez

rdečega. Kodiranje barvnih kanalov je opisano v poglavju 3.4. Ko sta oba pogleda obdelana in prikazujeta vsak svoje kanale, moramo sliko združiti na takšen način, da se pogled levega očesa v prosojnosti naloži v plasti čez pogled desnega očesa. Pri takšnem načinu združevanja se uporabi širina širše in višina višje slike. Ker je združevanje relativno in ne vidimo vsi enako, smo uporabniku ponudili tudi možnost ročnega prilagajanja zgornje slike, saj tako pustimo večjo možnost fleksibilnosti in izboljševanja slike.

Za karseda najboljši pogled smo pri prikazovanju anaglifske slike uporabili paralakso, ki prikaže sliko v horizontalnem zamiku. Nastavili smo privzeto vrednost paralakse na 64 pikslov, ki nam prikaže anaglifske sliko z najboljšim možnim globinskim učinkom. Paralakso lahko uporabnik prilagaja v urejevalniku slik.

4.5 Zajem slike z dvema napravama

Primarni cilj izdelave aplikacije za diplomsko nalogo je bil izdelava zajema stereoskopske slike s pomočjo dveh mobilnih naprav, kjer se napravi med seboj brezžično sinhronizirata in vzpostavi komunikacijo za sočasno zajemanje slike tako, da dobimo sliko posneti iz dveh pogledov. Sliki se združita in izdela se stereoskopska slika, ki uporabniku ustvari skozi zato namenjena očala prostorski pogled na sliko.

V aplikaciji smo implementacijo izdelali tako, da uporabniku nudi čim bolj enostaven način za uporabo ter nastavitve aplikacije. Omejitev za zajem slike pri dveh telefonih je ta, da moramo imeti aplikacijo nameščeno na obeh telefonih, saj mobilni napravi komunicirata preko aplikacije. Ker je tehnologija Bluetooth brezžična in za naše potrebe dovolj zanesljiva, je primerna za brezžično komunikacijo med mobilnimi aplikacijami. Komunikacija služi sinhronizaciji ter pošiljanju ukazov med mobilnimi napravami za sočasno zajemanje fotografij iz dveh pogledov, kjer podrejena naprava obravnava prejet ukaz ter nanj ustrezno reagira.

Izdelali smo aktivnost, ki nam prikaže seznam dosegljivih naprav z vključeno povezavo Bluetooth, s pomočjo katerega uporabniku omogočimo pregled nad napravami, s katerimi se lahko poveže. Da lahko začnejo mobilne naprave medsebojno komunicirati, jih moramo najprej povezati, pri čemer se vzpostavi komunikacijski kanal za komuniciranje in pošiljanje podatkov. Ker se podatki preko povezave Bluetooth pošiljajo v bajtih, smo morali sliko najprej pretvoriti v seznam bajtov in jo v takšni obliki poslati čez prenosni kanal. Na drugi strani glavna naprava, ki prejema sliko, poskrbi, da se slika v celoti prenese in sestavi. Slika se pošilja po kosih, zato mora glavna naprava sestavljati kose v celoto in po končanem

prenosu sliko shraniti v pomnilnik. S shranjevanjem zagotovimo, da sta fotografiji obeh pogledov na eni napravi za nadaljnje obdelovanje in združevanje v anaglifsko sliko.

Povezovanje in pošiljanje teče vzporedno v svoji niti zaradi bolj stabilnega in nemotenega delovanja. Povezava Bluetooth omogoča pošiljanje večjih podatkov in je tako primerna tudi za pošiljanje slik z ene naprave na drugo. Zaradi dolgotrajnih operacij, kot je pošiljanje slik, bi brez vzporednega delovanja mobilna aplikacija blokirala druge aktivnosti in postala neodzivna. Da se izognemo tej težavi in neprijetni uporabniški izkušnji, uporabimo ločeno nit in omogočimo upravljanje in nadzor nad aplikacijo tudi med daljšimi prenosi.

Zaradi brezžične povezave in prenašanja ukazov preko povezave Bluetooth prihaja do manjšega zamika zaradi hitrosti prenašanja podatkov ter različnih odzivnosti mobilnih naprav. Zato takšen način omogoča fotografiranje le statičnih slik, medtem ko je zajem dinamičnih predmetov lahko problematičen, ker mobilni napravi zajameta fotografijo s prevelikim zamikom.

4.5.1 Zajem slike z eno napravo

Ker uporabniki nimajo vedno možnosti uporabe dveh mobilnih naprav, smo v aplikaciji vključili tudi možnost zajema stereoskopske slike s samo eno mobilno napravo. Naprava uporablja svojo kamero za zajem slike, pri kateri se ob prvem zajemu ustvari prekrivanje za lažji in natančnejši zajem druge slike. Prekrivanje nam zelo olajša, da lahko kamero že sami poravnamo glede na prejšnjo sliko in tako ustvarimo zelo natančno sliko. Problem takšnega zajema je, da ni prilagojeno za primikajoče objekte in da mora imeti uporabnik nekaj napotkov glede prestavljanja mobilne naprave pri zajemanju drugega pogleda.

Zajemanja z eno napravo smo se lotili tako, da smo naredili pogled kamere, ki zajame posnetek, ko se dotaknemo zaslona. Ob zajemu slike se ta naloži kot plast v pogledu kamere ter tako uporabniku doda informacijo, na katero se orientira in uravnava drugo sliko. Plast smo implementirali tako, da smo definirali razred `LayoutInflater`, ki poskrbi, da se stilski predloga prenese v pogled, ki ga dodamo v obliki parametrov k obstoječemu pogledu. Predloga, ki smo jo naredili, vsebuje vsebovalnik za sliko, kamor se prenese prva fotografija, ki jo raztegnemo čez celoten zaslon.

4.6 Problemi pri implementaciji

Ker pri razvoju navadno naletimo na težave in probleme, kjer določeni deli kode ne delujejo tako, kot bi morali, smo tudi pri izdelavi diplomske naloge naleteli na nekaj problemov, ki jih v nadaljevanju podrobneje opišemo ter razčlenimo.

4.6.1 Pomanjkanje pomnilnika za delo z bitnimi slikami

S problemom smo se srečali, ko je aplikacija za delo uporabljala bitne slike večjih velikosti in je zmanjkalo pomnilnika, namenjenega za shranjevanje. Mobilna naprava hrani vse bitne slike v kopici omejenega prostora, ki ga lahko dokaj hitro zapolnimo. Vsakič, ko naredimo novo instanco razreda Bitmap, se v pomnilnik naloži polna velikost slike. Torej če imamo 24-bitno sliko velikosti 100 x 100 pikslov, to predstavlja 240.000 bitov. Ker ima en bajt 8 bitov, za shranjevanje v pomnilniku porabimo 30 KB prostora. Naše posnete fotografije so bile velikosti 1280 x 960 pikslov in najmanjša velikost kopice je bila 4 MB, kar pomeni, da v primeru, ko naša bitna slika zasede 3.5MB prostora, ob nalaganju nove bitne slike takoj zapolnimo kopico.

Napako smo odpravili tako, da smo povečali namenjen pomnilniški prostor za delo z bitnimi slikami. To smo storili tako, da smo v datoteki manifest povedali, da želimo uporabljati večjo kopico in ob kreiranju bitnih slik zmanjšamo kvaliteto ter se tako za naše potrebe izognemo problemu. Druga možnost, da se rešimo problem, je ta, da za vsako bitno sliko, ki je ne uporabljamo več, pokličemo smetarja (Garbage Collection), ki ob prvi priliki počisti našo bitno sliko iz pomnilnika. Da počistimo pomnilnik, ki ga zaseda instanca razreda Bitmap, kličemo metodo recycle, vendar v našem primeru to ni bilo najbolj primerno, saj dinamično nalagamo bitne slike in delamo z njimi, ta rešitev pa je bolj primerna za statične, ne spreminjajoče se slike.

4.6.2 Prenašanje slik preko povezave Bluetooth

Najprej smo implementirali komunikacijo med telefonoma preko povezave Bluetooth. Komunikacija deluje tako, da pošiljamo krajše nize znakov, pretvorjene v bajte, čez komunikacijski kanal. Problem se je pojavil, ko smo na isti način želeli pošiljati sliko, pretvorjeno v bajte, saj nismo prejeli le enega bloka bajtov kot pri nizu. Problem pri prenašanju slike preko povezave Bluetooth se je pojavil, ko smo želeli ugotoviti, kdaj se slika prenese do konca, oziroma kdaj so vsi bajti slike preneseni na mobilno napravo. Problem nastane, ker slika prenaša kose, ki se pri prejemniku sestavljajo v celoto. Kljub temu, da se slika mogoče že v celoti prenesla, še vedno prejemamo podatke z druge naprave in ne vemo, da se je prenos slike že zaključil. Bajt slike moramo shraniti takoj ob zaključku njenega

prenosa, da se drugi bajti ne shranjujejo pod sliko in je ne popačijo, saj ostane komunikacijski kanal vzpostavljen in se preko njega še vedno prenašajo tudi drugi podatki.

Problema smo se lotili tako, da smo pred pošiljanjem sliko stisnili v format JPEG s prilagojeno kvaliteto, pri prenosu pa smo preverili, ali smo že prenesli tudi konec datoteke. Po specifikacijah standarda JPEG [5] je datoteka s sliko razdeljena na več segmentov, zadnji segment pa je EOI, ki določa konec slike. Ta segment označujeta bajta 0xFF in 0xD9. Zaradi bajtov, ki določajo konec slike, smo lahko preverjali, kdaj se prenašanje konča, tako, da smo ob vsakem prenosu bloka pregledali, če je predzadnji bajt naše slike 0xFF in zadnji 0xD9. Takoj ko se prenesejo zadnji bajti slike, shranimo fotografijo, da imamo obe fotografiji na eni mobilni napravi. Nato zaženemo novo aktivnost za urejanje slik, kjer združujemo fotografiji v anaglifsko sliko.

4.6.3 Prikaz napačno obrnjene slike

Ko smo spisali kodo za zajemanje slike s pomočjo mobilne kamere, smo ugotovili, da se v pogledu kamere prikazuje slika, ki je za 90 stopinj obrnjena. Problem je v nekaterih napravah, ki prikazujejo in shranjujejo sliko v ležečem položaju. Ko smo odpravili problem prikazovanja pogleda kamere, smo ugotovili tudi, da se ob shranjevanju slika prav tako shrani v ležečem položaju, zato smo morali poskrbeti tudi za obračanje slike ob shranjevanju. Odpravljanje napake z obrnjeno sliko smo rešili tako, da smo ob ustvarjanju pogleda kamere nastavili parametre za rotacijo za 90 stopinj. Ker z rotacijo prikazovalnika kamere nismo obrnili naše slike ampak naš vsebovalnik za pogled, smo ob shranjevanju fotografije poskrbeti za obračanje.

4.6.4 Shranjevanje slike

Problem se je pojavil, ko smo želeli shraniti celotno sliko z obema plastema združevanja in ohraniti prosojnost na zgornji sliki. Razred Canvas hrani v pomnilniku vse akcije in izrise, zato je bila za nas to ena glavnih iztočnic, na osnovi katere smo tudi implementirali rešitev. Ker imamo za shranjevanje svoj gumb, se je pojavil tudi problem, da ko smo stisnili na gumb za shranjevanje, se je naša slika premaknila na koordinate gumba. To smo rešili s shranjevanjem prejšnjih koordinat, kjer smo ob kliku prepisali nove koordinate s prejšnjimi.

4.7 Testiranje aplikacije

Testiranje je eden ključnih procesov pri izdelavi kvalitetne in učinkovite mobilne aplikacije, zato smo se področju testiranja še posebej posvetili. Testiranje je potekalo v dveh delih. Med samim razvojem so se testirali deli kode in uporabnost aplikacije, kjer smo iz prve roke ugotovili, ali se pojavljajo kakšne napake. Drugi del testiranja je potekal po končanem razvoju aplikacije in je bil namenjen testiranju uporabniškega vmesnika in odkrivanju napak. Testni uporabniki so imeli zelo različne izkušnje z uporabo operacijskega sistema Android in aplikacij, tako da smo pokrili čim širši spekter uporabnikov z različnimi znanji in izkušnjami. Uporabniki so sporočili napake, opažanja in nasvete, ki so jih odkrili pri uporabi mobilne aplikacije. Na podlagi odzivov uporabnikov smo odpravili napake in mobilno aplikacijo ustrezno nadgradili.

Za testiranje mobilne aplikacije smo uporabili tri različne mobilne naprave: Samsung Galaxy Note 3, Samsung Galaxy Trend Plus in Samsung Galaxy S3 Mini. Naprave se med seboj razlikujejo po velikosti ekrana, kvaliteti zajemanja slik, velikosti slik in še veliko drugih stvari, tako da smo dobili idealne pogoje za pokritje večjega dela mobilnih naprav z operacijskim sistemom Android. Testiranje je namenjeno odkrivanju napak zaradi različnih komponent in nastavitev na mobilnih napravah. Na mobilnih napravah z različnimi različicami operacijskega sistema Android smo opazovali različno obnašanje postavitve na zaslonu in sočasno odpravljali napake v delovanju zaradi različnosti naprav.

Samsung Galaxy Note 3 [17] je pametni mobilni telefon, ki za svoje delovanje uporablja Android operacijski sistem. Razvilo ga je podjetje Samsung Electronics in so jih prodali več kot v 5 milijonov v prvem mesecu ter več kot 10 milijonov samo v drugem mesecu. Iz teh podatkov lahko zaključimo, da je ta mobilna naprava primerna za testiranje aplikacije, saj jo uporablja velik delež uporabnikov. Za potrebe diplomske naloge nas zanima kamera, ki ima 13 milijonov točk in ločljivost 4128 x 3096 slikovnih pik, kar pomeni, da s pomočjo kvalitetne kamere dobimo osnovo za primerjavo z drugimi testnimi napravami, ki imajo slabšo kvaliteto in resolucijo. Na mobilni napravi je nameščen operacijski sistem Android 4.4.2.

Mobilna naprava Samsung Galaxy Trend Plus [19] ima nameščen operacijski sistem Android 4.2.2 ter kamero s 5 milijoni točk ločljivosti 2592 x 1944, kar proti Samsung Galaxy Note 3 predstavlja precejšno razliko v kvaliteti in velikosti. S pomočjo te naprave smo lahko testirali, kako se mobilna aplikacija obnaša ob združevanju fotografij, ki so različnih ločljivosti.

Samsung Galaxy S3 Mini [18] ima vgrajen fotoaparatus s 5 milijoni točk ločljivosti 2592 x 1944. Ker ima naprava zelo podobno kamero kot Samsung Galaxy Trend Plus, je bil namen

vklučitve te naprave v testiranje, da testiramo aplikacijo in obnašanje na starejših različicah operacijskega sistema Android. Naprava uporablja operacijski sistem Android 4.1.

Testni uporabniki so mobilno aplikacijo testirali tako, da je vsak uporabnik zajel 20 testnih posnetkov, kjer so za vsak način zajemanja naredili 10 fotografij. Uporabniki so skozi delo z mobilno aplikacijo spoznavali delovanje in si beležili pomanjkljivosti in težave. Ker niso imeli vsi testni uporabniki svojih mobilnih naprav z operacijskim sistemom Android, smo vsem ponudili naše mobilne naprave. Prvi in drugi testni uporabnik sta dobila Samsung Galaxy Note 3 in Samsung Galaxy Trend Plus, kjer sta testirala uporabniški vmesnik in poročala o morebitnih težavah, ki sta jih imela z zajemanjem in prikazovanjem stereoskopskih slik. Poročala sta o razlikah pri zajemu slike z dvema mobilnima napravama, kjer prihaja do razlike med stereoskopskima slikama zaradi različne leče kamere. Ker se kameri razlikujeta, dobimo različen prikaz barve in drugačno povečavo na objekt. Tretji in četrti testni uporabnik sta dobila Samsung Galaxy Note 3 in Samsung Galaxy S3 Mini, kjer sta imela isto nalogo, kot prva dva testna uporabnika, vendar sta zaradi boljšega poznavanja operacijskega sistema Android bila pozorna tudi na napake, ki se lahko pojavijo zaradi različnih različic operacijskega sistema Android. Mobilna aplikacija se na vseh testiranih različicah operacijskega sistema Android obnaša enako. Zadnja dva testna uporabnika s slabšim poznavanjem operacijskega sistema Android pa sta testirala delovanje in uporabniški vmesnik na napravah Samsung Galaxy Trend Plus in Samsung Galaxy S3 Mini, kjer nista imela težav pri zajemanju slik, imela pa sta jih z uporabo uporabniškega vmesnika. Težava je nastala, ko sta želela vzpostaviti povezavo Bluetooth z drugo napravo, saj kadar naprava ni povezana, vmesnik ne ponudi seznama dosegljivih naprav samodejno, ampak ga moramo odpreti sami. Isto težavo so imeli ostali testni uporabniki, vendar so sami ugotovili način, da vzpostavijo povezavo. Uporabniški vmesnik pa, razen omenjene težave, testnim uporabnikom ni predstavljal težav in so ga upravljali samostojno, brez dodatne pomoči. Vsi uporabniki so bili po odpravljanju težav zadovoljni s končnim prikazom stereoskopske slike.

Na podlagi rezultatov testiranja smo ugotovili, da zajemanje stereoskopske slike s pomočjo ene naprave na vseh napravah deluje brez težav. Zaradi neodvisnosti od drugih naprav je takšen način zajemanja uporabnikom bolj prijazen kot zajemanje z dvema napravama, ker za delovanje potrebujejo samo svojo mobilno napravo in ker lažje poravnajo stereoskopske slike, kar se posledično kaže pri prikazu kvalitetnejših anaglifov.

Poglavje 5 Sklepne ugotovitve

V diplomskem delu smo opisali način zajemanja in prikazovanja stereogramov in opisali implementacijo in tehnologije naše mobilne aplikacije. Kljub vsem zapletom in težavam tekom razvoja smo naredili mobilno aplikacijo, ki vključuje vse zastavljene specifikacije in ciljne funkcionalnosti. Skozi načrtovanje in razvoj smo pridobili nova znanja uporabljenih tehnologij in nadgradili znanje razvoja mobilnih aplikacij za operacijski sistem Android. Podrobneje smo se spoznali s sestavljanjem anaglifске slike in tako z vsemi pridobljenimi znanji naredili mobilno aplikacijo, ki je pripravljena za zajemanje prostorske slike.

Pokazali smo način združevanja stereogramov v anaglifsko sliko ter predstavili tehnologije in možnosti zajemanja stereoskopske slike z mobilnimi napravami. Spoznali smo se s tehnikami stereoskopije in delovanjem človeškega vida ob gledanju prostorskih slik. Opisali smo tudi način implementacije združevanja v anaglifsko sliko, kar lahko uporabimo tudi pri izdelavi drugih aplikacij. Takšen način združevanja deluje samostojno, brez navezovanja na druge aktivnosti. Podobno tudi način komunikacije med mobilnimi napravami pomaga pri idejah, ki jih lahko uporabimo za komunikacijo med vsemi napravami s podporo tehnologije Bluetooth, kjer bi logiko delovanja lahko uporabili za katero drugo tehnologijo.

Da bi bila aplikacija boljša in bolj kvalitetna, bi lahko izpopolnili kar nekaj stvari. Lahko bi nadgradili uporabniški vmesnik tako, da bi oblikovanje poenotili po vseh aktivnostih in s primernejšo grafično podobo aplikacije pripomogli k boljši uporabniški izkušnji.

Izboljšala bi se lahko kvaliteta slik, kjer bi se morale slike zajemati v večji resoluciji, vendar se pojavi problem zaradi pomanjkanja pomnilnika pri shranjevanju bitnih slik. Če bi želeli aplikacijo nadgraditi na takšen način, bi morali prilagoditi kodo za sprotno sproščanje pomnilnika, ki je rezerviran za shranjevanje bitnih slik, ali pa bi uporabili drug primeren format za delo z slikami.

Pri zajemanju stereoskopske slike z dvema napravama bi se lahko ukvarjali s hitrejšo odzivnostjo slikanja druge naprave ter tako izboljšali slikanje primikajočih objektov. Prvi način, s katerim bi lahko aplikacijo za zajemanje dinamičnih objektov izboljšali, je ta, da bi predhodno določili, katera naprava je glavna in katera podrejena, in bi zajem občutno pohitрили, saj pošiljanje signala na podrejeno napravo ne bi bilo več potrebno. Pri taki rešitvi bi

moral uporabnik pritisniti gumb za slikanje na obeh telefonih istočasno, kjer bi podrejena naprava samodejno ob zajemu poslala sliko glavni napravi. Drugi način pa bi upošteval zamik, ki ga tehnologija Bluetooth porabi v povprečju. Na podlagi sinhronizacije bi mobilni napravi na začetku s testnimi ukazi ugotovili zamik, s katerim bo glavna naprava zajela sliko. Kadar bi za zakasnitev na glavni napravi upoštevali izmerjen čas, bi zajeli oba pogleda v istem času ter tako zagotovili funkcionalnost zajemanja dinamičnih slik.

Nadgradili bi lahko galerijo in uporabnikom omogočili izbiro načina prikaza slik. Nadgradili bi tudi urejanje slik, kjer bi vključili urejanje po zelenem vrstnem redu in slikam dodali dodatno informacijo o lokaciji posnetka. Funkcija razvrščanja slik v direktorije bi prinesla dodatno zadovoljstvo, kjer bi poljubno dodajali fotografije v svoje direktorije ter izboljšali preglednost.

Združevanje slik v anaglifske slike lahko nadgradimo z umetno inteligenco, kjer bi na slikah iskali točke interesa. Točke interesa nam predstavljajo zanimivosti na fotografiji, na podlagi katerih bi lahko naredili algoritem za samodejno popolno združevanje z upoštevanjem paralakse. Ogrodje OpenCL, ki obstaja tudi v različici za operacijskim sistemom Android, vsebuje knjižnice za delo z računalniškim vidom. Z algoritmom SIFT lahko izračunamo točke interesa obeh slik ter sliko prilagajamo tako, da se pokriva čim več točk.

Operacijski sistem Android je najbolj razširjen med uporabniki, vendar število uporabnikov operacijskih sistemov iOS, Blackberry in Windows Phone prav tako raste. Če bi želeli, da našo aplikacijo uporabljajo tudi na drugih operacijskih sistemih, bi jo morali implementirati tudi na drugih platformah, ali pa uporabiti pretvornike, ki poskrbijo, da naša aplikacija deluje tudi na drugih platformah. Pretvorba aplikacij ne deluje vedno tako, kot bi si želeli, zato je najboljši način, da izdelamo namensko mobilno aplikacijo za določeno platformo.

Vsaka omenjena nadgradnja mobilne aplikacije, ki je predstavljena v diplomskem delu, bi pripomogla k boljši uporabniški izkušnji. Z zadovoljstvom uporabnikov povečamo možnost za uspešnost aplikacije in tako pripomoremo k temu, da uporabniki uporabljajo našo mobilno aplikacijo. Z vsemi zastavljenimi cilji v diplomskem delu smo dosegli, da uporabniku ponujamo kvalitetno mobilno aplikacijo, ki jo lahko uporablja za zajem prostorskih slik.

Literatura

- [1] P. Bhagwat, "Bluetooth: technology for short-range wireless apps", Internet Computing, IEEE, št. 3, zv. 5, str. 96-103, 2001.
- [2] E. Burnette, "Hello, Android: Introducing Google's Mobile Development Platform", Raleigh, North Carolina, 2009.
- [3] A. Higgott in T. Wray, "Camera Constructs : Photography, Architecture and the Modern City", Farnham, Surrey, England, 2012.
- [4] K. Keresteš, "Making the anaglyph map", Geodetski vestnik, zv. 44, št. 1 in 2, 2000.
- [5] W. B. Pennebaker in J. L. Mitchell, "JPEG Still Image Data Compression Standard", 1. izd, Norwell, USA, 1992.
- [6] T. Shibata, "Head mounted display", Displays, št. 23, Tokyo, Japan, 2002, str. 57.
- [7] A. J. Woods in C. R. Harris, "Comparing levels of crosstalk with red/cyan, blue/yellow, and green/magenta anaglyph 3D glasses", Stereoscopic Displays and Applications, št. 21, str. 0Q1-0Q12, 2010.
- [8] (2014) Anaglyphic 3D. [Online]. Dosegljivo:
http://www.arachnoid.com/raytracing/anaglyphic_3d.html.
- [9] (2014) Android Bitmaps [Online]. Dosegljivo:
<http://www.informit.com/articles/article.aspx?p=2143148&seqNum=2>.
- [10] (2014) Android SDK. [Online]. Dosegljivo:
<http://developer.android.com/sdk/index.html>.
- [11] (2014) Android View [Online]. Dosegljivo:
<http://developer.android.com/reference/android/view/View.html>.

- [12] (2014) App Manifest [Online].
Dosegljivo:<http://developer.android.com/guide/topics/manifest/manifest-intro.html>.
- [13] (2014) Camera [Online]. Dosegljivo:
<http://developer.android.com/guide/topics/media/camera.html>.
- [14] (2014) Guide to Shooting Stereoscopic 3D. [Online]. Dosegljivo:
<http://www.dashwood3d.com/blog/beginners-guide-to-shooting-stereoscopic-3d/>.
- [15] (2014) John Bercovitz. [Online]. Dosegljivo:
<http://nzphoto.tripod.com/stereo/3dtake/newversionberk.html>.
- [16] (2014) Layouts [Online]. Dosegljivo:
<http://developer.android.com/guide/topics/ui/declaring-layout.html>.
- [17] (2014) Samsung Galaxy Note 3. [Online]. Dosegljivo:
http://www.gsmarena.com/samsung_galaxy_note_3-5665.php.
- [18] (2014) Samsung Galaxy S3 Mini. [Online]. Dosegljivo:
http://www.gsmarena.com/samsung_i8190_galaxy_s_iii_mini-5033.php.
- [19] (2014) Samsung Galaxy Trend Plus. [Online]. Dosegljivo:
http://www.gsmarena.com/samsung_galaxy_s_duos_2_s7582-5876.php.
- [20] (2014) Smartphone OS Market Share. [Online]. Dosegljivo:
<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>.